NASA-TM-111504

Question-based Acquisition of Conceptual Indices for Multimedia Design Documentation

CATHERINE BAUDIN
SMADAR KEDAR
JODY GEVINS-UNDERWOOD
RECOM TECHNOLOGIES, INC.
VINOD BAYA
STANFORD UNIVERSITY

ARTIFICIAL INTELLIGENCE RESEARCH BRANCH
NASA AMES RESEARCH CENTER
MAIL STOP 269-2
MOFFETT FIELD, CA 94035-1000

NASA Ames Research Center
Artificial Intelligence Research Branch
Technical Report FIA-93-22
July, 1993

ulcu(indice,) Tj 4

conceptual indices, for instance by interpreting sentences in a text [Mauldin 91] [Tong et al. 89]. On the other hand, the creation of conceptual indices by human indexers is a labor intensive task that is difficult to perform exhaustively. This is particularly true for a large volume of documentation where concepts are closely interrelated, as is the case for technical documents that describe the operation, diagnosis or design of complex artifacts.

2. Question-based Acquisition of Conceptual Indices

Our approach is to use a *conceptual query language* plus feedback from the user on the relevance of the documents retrieved in response to a query, to incrementally acquire new conceptual indices for that document. The user formulates a query to the system. If no document description exactly matches the query, the system approximates the retrieval and prompts the user for feedback on the relevance of the references retrieved. If a reference is confirmed, the query is turned into a new index. This extends *relevance feedback* techniques [Salton et al. 68][Salton et al. 88] to the acquisition of *conceptual indices.*

This approach uses a *question-based indexing* paradigm [Osgood et al. 91][Schank 91][Mabogunje 90] where the query language and the indexing language have the same structure and use the same vocabulary. The assumption is that the questions asked by users indicate the objects and relationships that are relevant to describe the content of the documents at a conceptual level appropriate for a class of users. However, in order to use the queries to acquire new indices the following conditions must be met by the query language:

1. Reusability: The query language must be general enough to create indices that will match a class of queries.
2. Relevance: The query language must be able to describe

3. Background

We developed Dedal, an information retrieval system that uses conceptual indexing to represent the content of multimedia text, graphics and videotaped design information. Dedal is currently applied to documents of mechanical engineering design. It is an interface to records such as meeting summaries, pages of a designer's notebook, technical reports, CAD drawings and videotaped conversations between designers.

3.1 Conceptual Language to Query and Index Design Information

Based on studies of the information seeking behavior of designers conducted at Stanford's Center for Design Research and NASA Ames [Baya et al. 92], we identified a language to describe and query design information [Baudin et al. 92a]. This language combines concepts from a model of the artifact being designed with a task vocabulary representing the classes of design topics usually covered by design documents. For instance, "function," "operation," or "alternative" are topics of the task vocabulary.

A conceptual index can be seen as a structured entity made of two parts: the *body* of the index which represents the content of a piece of information and the *reference* part that point to a region in a document. In Dedal the body of an index has the following form: <topic T subject S level of detail L medium M> where S is a list of subjects from a domain model and T, L and M are member of the task vocabulary. The reference part of an index contains a pointer to the *record* and *segment* corresponding to the starting location of the information in a document (e.g. document name and page number or video counter). A segment of information is described by several conceptual indices, each of which partially describing its content.

For instance: "The inner hub holds the steel friction

3.2 The domain model

In the mechanical engineering design domain, the model includes a representation of the artifact structure, some aspects of its function, the main decision points and alternatives considered. It also includes concepts that are part of the problem but external to the device representation. The main relations in the model are *isa*, *part-of*, *attribute-of*, and *depends-on* (see Figure 1). The *isa* and *part-of/ attribute-of* hierarchies are used by Dedal to compare a query with a given index. For instance in Figure 1, given that *metal-disk* is part of the *disk-stack* the pattern: "function of metal-disk" will be considered more specific than the pattern "function of disk-stack". In the same way, the subject: "resistive-force of disk-stack" is more specific than the subject "disk-stack".

3.3 Retrieval strategy

The retrieval module takes a query from the user as input, matches the question to the set of conceptual indices and returns an ordered list of references related to the question. The retrieval proceeds in two steps: (1) exact match: find the indices that exactly match the query and return the associated list of references. If the exact match fails: (2) approximate match: activate the *proximity retrieval heuristics*.

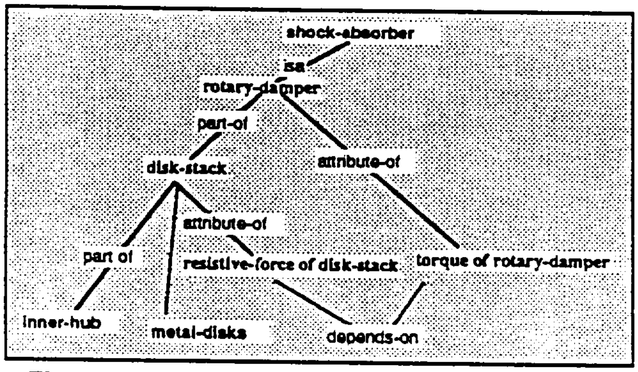

Figure 1: Objects and relations in the domain model

Dedal currently uses fourteen proximity retrieval heuristics to find related answers to a question. For instance, segments described by concepts like "*decision* for lever material" and "*alternative* for lever material" are likely to be located in nearby regions of the documentation. The heuristics are described in detail in [Baudin et al. 92b].

Each retrieval step returns a list of references ordered according to a set of priority criteria. The user selects a reference and if the document is on line, goes to the corresponding segment of information (using the hypertext facility that supports the text and graphics documents). A user dissatisfied with the references retrieved can request more information and force Dedal to resume its search and retrieve other references.

4. Index Acquisition in Dedal

Dedal acquires a new index in two phases: (1) an index creation phase, and (2) an index refinement phase.

4.1. Index Creation

Figure 2 illustrates with an example how Dedal acquires a new index. Given a user query and feedback from the user on the relevance of the documents retrieved. The index creation phase goes through the following steps:

1. Query formulation: The user's question in English is "what is the function of the hub?". After the user selects the subject *inner-hub* from the domain model and the topic *function* from the task vocabulary, the corresponding query in Dedal is: < topic: function of subject: inner-hub>(In the following paragraphs we will use a shortened syntax for queries where the words topic and subjects are omitted and where domain concepts are indicated in bold).

2. Query-Index mapping: Dedal tries to find an index that exactly matches the query. In this case, it does not find an exact match and applies a proximity heuristic to guess where the required information may be located. The heuristic states that any information describing how a mechanism works might also describe the function of its parts. In this case, given that *inner-hub* is a subpart of the *disk-stack* mechanism, Dedal matches the query "function of **inner-hub**" with two indices I1 and I2 pointing to two information regions describing the "operation of **disk-stack**".

3. Relevance Feedback: The user looks at the two references retrieved, finds that the reference pointed to by the index I2 (page 12 in the record report-333) describes the function of the inner hub while the document associated with index I1 does not. The user rates the reference I2 as relevant.

4. Index Acquisition: The query: "function of inner-hub" is more specific (see section 3) than the index "operation of disk-stack". In this case Dedal creates a new index I3. The system now knows that page 12 of report-333 explicitly describes the function of the inner-hub.

Each time a reference is retrieved by the approximate match and is relevant, Dedal attaches the reference of the selected index to the query, turning the query into a new index (as shown in step 4 in figure 2). In addition, the procedure records the type of inference that relates each subject of the new question to the subject of the matching index. There are four types of inferences: *identity*, *specialization*, *generalization* and *extension*. These inferences determine the type of the subjects associated with the new index created.

The type of a new subject is *identity* if this subject is identical to a subject of the matching index. The type of the new subject is a *specialization* if it is related to a subject of the matching index by a subpart or isa relationship, or if its value depends-on the value of the matching subject. The type of subject is a *generalization* if the matching subject is related to the new subject by a subpart or isa relation, or if its value depends-on the value of the new subject. The type of the new subject is an *extension* if it has no relations with any of the matching subjects. Finally if an index is defined manually by a user, the type of its subjects is: *human-indexer*.

For instance, if the query : "relation between **solenoid** and lever" matches the index: <topic: operation, subject: solenoid, reference: (meeting-10/2/91, 12)>, the new index will be: <topic: relation, subject1: solenoid and subject2: lever, reference(meeting-10/2/91, 12)>, where type(subject1) = identity and type(subject2)= extension.

When a query is matched to a new index created by Dedal, the type of each subject is taken into account in the determination of the ordering coefficient. The greatest confidence is attached to subjects in the following order: human-indexer, identity, specialization, generalization and extension. This means that there is high confidence in a new index created by a human while little confidence if the index is overgeneralized or provides an unrelated reference.

4.2 Index Refinement

Two factors may impact the ability of an acquired index to accurately describe the associated information:

(1) *incompleteness of the domain model:* If the model is missing the particular subject the user is interested in and the user selects a related subject, the approximate match might still retrieve a relevant document. In this case the user query does not exactly describe the information required by the user and the resulting index will be inaccurate;

(2) *multiple subject problem:* when a query involves several subjects from the model, the user might feel satisfied with a document that refers to a *subset* of these subjects. For instance, if the query is of the form "relation between **outer-cage, solenoid** and **lever**" the user might feel satisfied with a reference which only describes the relation between outer-cage and solenoid, the third argument: lever will then incorrectly describes the content of the referenced document.

The index refinement phase keeps track of the relevance of each subject in the newly acquired indices. Each time a query Q matches an acquired index I, and a subject Sq of Q is related to the subject Si of the index I (where related means either is the identity, a specialization, a generalization or an extension), the following procedure is activated: if the corresponding reference is relevant, the success rate of Si is incremented. If the reference retrieved is irrelevant, the failure rate of Si is incremented. The idea is that after some time, the indices that are suspect (whose failure rate is above a certain threshold) will be presented to a human indexer who will decide what indices should be maintained or deleted and what subjects should be dropped from the index.

For example, if the question is "what component interacts with the lever?", the corresponding query : < relation (between) lever and SX> (where SX is a variable) matches the body of the index I: < relation (between) **solenoid, lever, shaft** >. If the match is rated by the user as relevant, the coefficient of success of the subject lever in index I will be reinforced. If the user indicates that the reference retrieved is not relevant, the coefficient of failure of subject lever in I will be reinforced. Eventually if the index I fails to match any query about lever, the subject lever will be dropped.

1. User: what is the function of the inner-hub?
2. Exact match: Find a conceptual index with topic = function and subject = inner-hub
-> the retrieval fail
Heuristic: If the query is function of X and X is a subpart of Y, look for documents that describe how Y works —> find indices with topic = operation and subject = Y.
From the domain model: inner-hub is part of disk-stack.
Approximate match: Find an index with topic = operation and subject = disk-stack
- ->Two indices are found:

I1: topic = operation, subject = disk-stack level-of-detail = conceptual in-record: meeting-12/2/90	I2: topic: operation, subject: disk-stack level-of-detail = detailed in-record: report-333	Body

```
index: EXP1-Q17
 in-record: DAMPER-ORD-WINTER-1990,
 in-segment: 23
 Topic: LOCATION of Subject: ARM
 created by rule: PART-OF
 from question: Q15
 from index G205:
   Topic: DESCRIPTION
      of Subject: ROTARY-DAMPER
```

(a)

```
index: EXP1-Q59
 in-record: DAMPER-ORD-SPRING-1990,
 in-segment: 12
 Topic: RELATION of Subjects:
          (SUSPENSION-SYSTEM, DAMPER)
 created by rule: OR-RULE
 from question: Q58
 from index G329:
   Topic: RELATION of Subject:
          (SUSPENSION-SYSTEM, CAR)
```

(b)

Figure 3: Two indices generated by Dedal

5. Experiments and Preliminary Results

In this section we report on experiments and preliminary results to evaluate the effectiveness of Dedal's index acquisition. Index Acquisition is considered *effective* by three criteria: *reusability*, *relevance* and *context independence* of the indices in future retrieval (see section 2 for a description of these criteria).

We conducted experiments where we observed mechanical engineers using Dedal to ask questions in the context of a modification of a shock absorber designed at Stanford's Center for Design Research for Ford Motor Corporation [Baudin et al. 92a]. The engineers rated the references retrieved by Dedal as relevant or irrelevant. In these experiments we considered three contextual factors: the *user*, the *problem* being solved, and the specific *goal* that motivates each query.

Experiment 1: In the first experiment a mechanical designer unfamiliar with the shock absorber design queried the system during redesign. In this experiment we measured the *relevance* and the *reusability* of the indices acquired within the same problem solving process. As the new indices were created, they were reused to answer slightly different questions. Out of 71 indices created, 13 were reused and out of those 70% were found relevant by the user. The main causes of irrelevance were the incompleteness of the model and the multiple subjects problem, where indices that involve relations among multiple subjects need more training to be refined (see Section 4.2).

Experiment 2: An expert designer used the system for a similar redesign task. In this study we observed how the indices created during experiment1 were reused in experiment2. This gave us an idea of the *reusability* and *relevance* of these indices, with a user of different design experience, and during the course of another problem solving process. In this experiment many questions were about the relation among multiple subject and we focused on the reusability and relevance of indices that have more than one subject. Each time a multiple subject index is reused, the success or failure coefficients of its subjects are updated by the system. We found that:

(1) The number of irrelevant new indices retrieved outweighed the number of new indices that were relevant, thus degrading the performance of the system. In this experiment 30 indices created during experiment1 were reused and out of these, 40% were relevant. As expected, this degradation was due to the multiple subject problem, mainly to the introduction of incorrect subject extensions.

(2) In the new indices, each incorrect subject was showing positive failure rates and no success rates. The new indices created were shown to another designer that confirmed the trend that the system recorded. This suggests that the accuracy of the indices created is improving and will lead to a better performance in future retrieval.

Figure 3b shows an index generated during the first experiment, in this index the subject "damper" is an incorrect extension of the original index "relation (between) suspension-system, car". The rating (not shown on the figure) of the subject "damper" showed a positive failure rating and no success rating after we conducted the second experiment.

Experiment 3: We presented the 71 indices created (see Figure 3) during experiment 1 along with the associated information regions to a designer familiar with the shock absorber documentation. The designer rated each of these indices as relevant or irrelevant depending on his appreciation of the ability of the index to describe (part of)the associated information. In this experiment, the designer reviewed the *relevance* and *context independence* of the indices created. The three contextual factors (user, problem and goals) in our experiment were removed: the designer was different from the users who conducted the experiments, he rated the indices independently of any problem solving task, and he had no access to the English version of the questions that motivated the queries. The designer rated 86% of the acquired indices as relevant. Here again the irrelevant indices acquired were due to the incompleteness of the domain model and the introduction of incorrect subject extensions in indices with more than one argument.

The three criteria, reusability, relevance and context independence, of the acquired indices don't give us a direct measure of the impact of these indices on the global retrieval performance in terms of the precision and recall of the retrieval[1]. However, when the newly acquired indices are reusable and relevant across contexts, the references associated with them can be retrieved through an exact match instead of an approximate match. Our assumption is that this provides better performance since the precision of the exact match retrieval is higher than the precision of the approximate retrieval [Baudin et al. 92b] and since the exact match will now retrieve more references. The intuition is that the user will see more relevant references sooner while more irrelevant inferences will be pruned form the first set of documents proposed to the user. For instance in the example discussed in Section 4.1, the system retrieved two references in response to the query "function of **inner-hub**", only one of this reference being relevant the precision of this retrieval was 50%. After Dedal acquired the new index I3 and the next time the same question is asked, only the relevant reference will be retrieved through an exact match in the first set of answers proposed to the user.

6 Future work

Performance evaluation: Our preliminary experimental results are mostly qualitative. They are useful in indicating the main features of the effectiveness of the index acquisition in terms of the reusability, relevance and context independence of the acquired indices. In order to have a more precise notion of the effectiveness of the method we plan to quantitatively evaluate the impact of these indices on the global performance of the system in terms of the gain in the precision and recall of Dedal.

Interactive modification of the domain model: The query language is designed to describe as much as possible the information required by the user. However, any language that uses concepts from a model is inherently incomplete. A missing domain subject forces the user to fall back on a related subject and is a source of inaccuracy in the use of queries for indexing purposes. One way of alleviating this problem is to allow the user to define new domain subjects when he cannot find a suitable concept in the model. We implemented a question formulation component that interacts with the user to understand how a new subject relates to the domain model and we plan to test this functionality with a user.

Definition of the domain model: Our conceptual query language is (1) task dependent: It is adapted to the type of questions that designers are interested in when they access design documents, and (2) is constrained by a domain model and requires this model to be built for each new design project. With respect to this method, an advantage of technical domains that relate to the operation, diagnostic or design of engineered artifacts, is that the scope of the domain model is usually well defined. For instance, in the engineering design domain a large part of technical documentation can be indexed using terms from a structural model (*part-of* hierarchy of components) of the designed artifact. The domain model becomes a design glossary whose terms are linked by different types of relations. Although model building might be considered a burden when compared to domain independent information retrieval systems, it is interesting to note that this type of "super-

RUBRIC (Tong 89) uses *evidential reasoning* and *natural language processing* techniques to infer the content of a text. For instance, an evidential rule can define which words and relations among words suggest a given concept. It is not clear, at this point, how much

Hayes, P., Pepper, J. "Towards An Integrated Maintenance Advisor" in Hypertext '89 Proceedings.

Mabogunje, A. "A conceptual framework for the development of a question based design methodology",